e-mail

Results for "((assertions failures)<in>metadata)"
Your search matched 1 of 1585504 documents.
A maximum of 100 results are displayed, 25 to a page, sorted by Relevance in Descending order.

» Search Options

View Session History

New Search

Modify Search

((assertions failures)<in>metadata) Search

Check to search only within this results set

Display Format: Citation Citation & Abstract

» Key

IEEE JNL	IEEE Journal or Magazine
IET JNL	IET Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IET CNF	IET Conference Proceeding
IEEE STD	IEEE Standard

view selected items Select All Deselect All

1. **Java model checking**
Park, D.Y.W.; Stern, U.; Skakkebaek, J.U.; Dill, D.L.;
Automated Software Engineering, 2000. Proceedings ASE 2000. The Fifteenth International Conference on
11-15 Sept. 2000 Page(s):253 - 256
Digital Object Identifier 10.1109/ASE.2000.873671
AbstractPlus | Full Text: PDF(320 KB) IEEE CNF
Rights and Permissions

Terms used **assertion failures** Found **21,803** of **203,282**

Sort results by: relevance
Display results: expanded form
Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next
Best 200 shown Relevance scale

1 Description logics of minimal knowledge and negation as failure
Francesco M. Donini, Daniele Nardi, Riccardo Rosati
April 2002 **ACM Transactions on Computational Logic (TOCL)**, Volume 3 Issue 2
Publisher: ACM Press
Full text available: pdf(471.25 KB) Additional Information: full citation, abstract, references, citings, index terms

We present description logics of minimal knowledge and negation as failure (MKNF-DLs), which augment description logics with modal operators interpreted according to Lifschitz's nonmonotonic logic MKNF. We show the usefulness of MKNF-DLs for a formal characterization of a wide variety of nonmonotonic features that are both commonly available inframe-based systems, and needed in the development of practical knowledge-based applications: defaults, integrity constraints, role, and concept closure. ...

Keywords: Description Logics, frame-based systems, nonmonotonic modal logics, tableau calculi

2 Service reasoning and monitoring: Associating assertions with business processes and monitoring their execution
Alexander Lazovik, Marco Aiello, Mike Papazoglou
November 2004 **Proceedings of the 2nd international conference on Service oriented computing ICSOC '04**
Publisher: ACM Press
Full text available: pdf(518.27 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Business processes that span organizational borders describe the interaction between multiple parties working towards a common objective. They also express business rules that govern the behavior of the process and account for expressing changes reflecting new business objectives and new market situations.

In our previous work we developed a service request language and support framework that allow users to formulate their requests against standard business processes. In this paper we ...

Keywords: management, monitoring, quality, service and AI computing, service delivery, theoretical frameworks for service representation and composition

3 Mining failures and bugs: Finding failure-inducing changes in java programs using change classification

Maximilian Stoerzer, Barbara G. Ryder, Xiaoxia Ren, Frank Tip

November 2006 **Proceedings of the 14th ACM SIGSOFT international symposium on Foundations of software engineering SIGSOFT '06/FSE-14**

Publisher: ACM Press

Full text available: pdf(275.16 KB) Additional Information: full citation, abstract, references, index terms

Testing and code editing are interleaved activities during program development. When tests fail unexpectedly, the changes that caused the failure(s) are not always easy to find. We explore how change classification can focus programmer attention on failure-inducing changes by automatically labeling changes *Red, Yellow,* or *Green*, indicating the likelihood that they have contributed to a test failure. We implemented our change classification tool *JUnit/CIA* as an ex-tension to ...

Keywords: change impact analysis, debugging, fault localization, testing, version control

4 A Sound Assertion Semantics for the Dependable Systems Evolution Verifying Compiler

Patrice Chalin

May 2007 **Proceedings of the 29th International Conference on Software Engineering ICSE '07**

Publisher: IEEE Computer Society

Full text available: pdf(426.15 KB) Additional Information: full citation, abstract

The Verifying Compiler (VC) project is a core component of the Dependable Systems Evolution Grand Challenge. The VC offers the promise of automatically proving that a program or component is correct, where correctness is defined by program assertions. While several VC prototypes exist, all adopt a semantics for assertions that is unsound. This paper presents a consolidation of VC requirements analysis activities that, in particular, brought us to ask targeted VC customers what kind of semantics ...

5 Assertional checking and symbolic execution: An effective combination for debugging

J. Mack Adams, James Armstrong, Melissa Smartt

January 1979 **Proceedings of the 1979 annual conference ACM 79**

Publisher: ACM Press

Full text available: pdf(496.24 KB) Additional Information: full citation, abstract, references, index terms

Software reliability will be no less a challenge in the 80's than in the previous decade but the basic research of the 70's can be applied to develop software tools to meet this challenge. In this paper one such application is described, namely the development of a debugging package based on two very active areas of research in the late 60's and throughout the 70's: research on proving assertions about programs [7,9,10,14,17] and on symbolic execution [3,11,12,13]. Combining these two areas ...

6 Asserting performance expectations

Jeffrey S. Vetter, Patrick H. Worley

November 2002 **Proceedings of the 2002 ACM/IEEE conference on Supercomputing Supercomputing '02**

Publisher: IEEE Computer Society Press

Full text available: pdf(351.59 KB) Additional Information: full citation, abstract, references, citings, index terms

Traditional techniques for performance analysis provide a means for extracting and analyzing raw performance information from applications. Users then compare this raw

data to their performance expectations for application constructs. This comparison can be tedious for the scale of today's architectures and software systems. To address this situation, we present a methodology and prototype that allows users to assert performance expectations explicitly in their source code using performance asse ...

7 Coverage based validation: On the evaluation of transactor-based verification for reusing TLM assertions and testbenches at RTL

Nicola Bombieri, Franco Fummi, Graziano Pravadelli

March 2006 **Proceedings of the conference on Design, automation and test in Europe: Proceedings DATE '06**

Publisher: European Design and Automation Association

Full text available: pdf(176.76 KB) Additional Information: full citation, abstract, references

Transaction level modeling (TLM) is becoming an usual practice for simplifying system-level design and architecture exploration. It allows the designers to focus on the functionality of the design, while abstracting away implementation details that will be added at lower abstraction levels. However, moving from transaction level to RTL requires to redefine TLM testbenches and assertions. Such a wasteful and error prone conversion can be avoided by adopting transactor-based verification (TBV). Ma ...

8 Is "sometime" sometimes better than "always"?: Intermittent assertions in proving program correctness

Zohar Manna, Richard Waldinger

October 1976 **Proceedings of the 2nd international conference on Software engineering ICSE '76**

Publisher: IEEE Computer Society Press

Full text available: pdf(838.53 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper explores a technique for proving the correctness and termination of programs simultaneously. This approach, which we call the intermittent-assertion method, involves documenting the program with assertions that must be true at some time when control is passing through the corresponding point, but that need not be true every time. The method, introduced by Knuth and further developed by Burstall, promises to provide a valuable complement to the more conventional m ...

9 Assertions and APL programming

Susan L. Gerhart

June 1975 **Proceedings of seventh international conference on APL APL '75**

Publisher: ACM Press

Full text available: pdf(576.09 KB) Additional Information: full citation, abstract, references, citings, index terms

Some difficulties in reading and writing APL programs are discussed. Assertions (important relations among and properties of variables) are suggested as a technique for improving reliability and readability of APL programs. It is shown how such assertions may be used for documentation, verification, optimization, and debugging. An implementation of run-time assertions in APLSV is given and illustrated by examples.

10 The notification based approach to implementing failure detectors in distributed systems

Jin Yang, Jiannong Cao, Weigang Wu, Corentin Travers

May 2006 **Proceedings of the 1st international conference on Scalable information systems InfoScale '06**

Publisher: ACM Press

Full text available: pdf(262.92 KB) Additional Information: full citation, abstract, references

Failure Detector (FD) is the fundamental component of fault tolerant computer systems. In recent years, many research works have been done on the study of QoS and implementation of FDs for distributed computing environments. Almost all of these works are based on the heartbeat approach (HBFD). In this paper, we propose a general model for implementing FDs which separates the processes to be monitored from the underlying running environment. We identify the potential problems of HBFD approach and ...

Keywords: QoS, failure detector, fault tolerance, heartbeat, performance evaluation

11 Session 6: test automation: Performance assertions for mobile devices

Raimondas Lencevicius, Edu Metz
July 2006 **Proceedings of the 2006 international symposium on Software testing and analysis ISSTA '06**
Publisher: ACM Press
Full text available: pdf(266.70 KB) Additional Information: full citation, abstract, references, index terms

Assertions have long been used to validate the functionality of software systems. Researchers and practitioners have extended them for validation of non-functional requirements, such as performance. This paper presents the implementation and application of the performance assertions in mobile device software. When applying performance assertions for such systems, we have discovered and resolved a number of issues in assertion specification, matching, and evaluation that were unresolved in previo ...

Keywords: assertions, mobile devices, performance

12 Performance assertion checking

Sharon E. Perl, William E. Weihl
December 1993 **ACM SIGOPS Operating Systems Review , Proceedings of the fourteenth ACM symposium on Operating systems principles SOSP '93**, Volume 27 Issue 5
Publisher: ACM Press
Full text available: pdf(1.16 MB) Additional Information: full citation, abstract, references, citings, index terms

Performance assertion checking is an approach to automating the testing of performance properties of complex systems. System designers write assertions that capture expectations for performance; these assertions are checked automatically against monitoring data to detect potential performance bugs. Automatically checking expectations allows a designer to test a wide range of performance properties as a system evolves: data that meets expectations can be discarded automatically, focusing a ...

13 Reports: A historical perspective on runtime assertion checking in software development

Lori A. Clarke, David S. Rosenblum
May 2006 **ACM SIGSOFT Software Engineering Notes**, Volume 31 Issue 3
Publisher: ACM Press
Full text available: pdf(473.68 KB) Additional Information: full citation, abstract, references, index terms

This report presents initial results in the area of software testing and analysis produced as part of the Software Engineering Impact Project. The report describes the historical development of runtime assertion checking, including a description of the origins of and significant features associated with assertion checking mechanisms, and initial findings about current industrial use. A future report will provide a more comprehensive assessment of development practice, for which we invite readers ...

14 Specification-based test oracles for reactive systems

Debra J. Richardson, Stephanie Leif Aha, T. Owen O'Malley

June 1992 **Proceedings of the 14th international conference on Software engineering ICSE '92**

Publisher: ACM Press

Full text available: pdf(1.74 MB) Additional Information: full citation, references, citings, index terms

15 Rx: treating bugs as allergies---a safe method to survive software failures

Feng Qin, Joseph Tucek, Jagadeesan Sundaresan, Yuanyuan Zhou

October 2005 **ACM SIGOPS Operating Systems Review , Proceedings of the twentieth ACM symposium on Operating systems principles SOSP '05**, Volume 39 Issue 5

Publisher: ACM Press

Full text available: pdf(245.29 KB) Additional Information: full citation, abstract, references, citings, index terms

Many applications demand availability. Unfortunately, software failures greatly reduce system availability. Prior work on surviving software failures suffers from one or more of the following limitations: Required application restructuring, inability to address deterministic software bugs, unsafe speculation on program execution, and long recovery time.This paper proposes an innovative *safe* technique, called Rx, which can quickly recover programs from many types of software bugs, both det ...

Keywords: availability, bug, reliability, software failure

16 Workshop on Architecting Dependable Systems (WADS): Failure modelling in software architecture design for safety

Weihang Wu, Tim Kelly

May 2005 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 2005 workshop on Architecting dependable systems WADS '05**, Volume 30 Issue 4

Publisher: ACM Press

Full text available: pdf(115.77 KB) Additional Information: full citation, abstract, references, index terms

In mission-critical industries, early feedback on the safety properties of a software system is critical and cost effective. This paper presents a compositional method for failure analysis of a system based on the proposed software architecture. This method is based upon the use of CSP as the failure modelling language and its associated tools as failure analysis. Preliminary findings from the application of this approach are also presented.

Keywords: CSP, failure modelling, safety analysis, software architectures

17 Bug isolation via remote program sampling

Ben Liblit, Alex Aiken, Alice X. Zheng, Michael I. Jordan

May 2003 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2003 conference on Programming language design and implementation PLDI '03**, Volume 38 Issue 5

Publisher: ACM Press

Full text available: pdf(258.37 KB) Additional Information: full citation, abstract, references, citings, index terms

We propose a low-overhead sampling infrastructure for gathering information from the executions experienced by a program's user community. Several example applications illustrate ways to use sampled instrumentation to isolate bugs. Assertion-dense code can

be transformed to share the cost of assertions among many users. Lacking assertions, broad guesses can be made about predicates that predict program errors and a process of elimination used to whittle these down to the true bug. Finally, even ...

Keywords: assertions, bug isolation, feature selection, logistic regression, random sampling, statistical debugging

18 A locking protocol for resource coordination in distributed databases

Daniel A. Menasce, Gerald J. Popek, Richard R. Muntz

June 1980 **ACM Transactions on Database Systems (TODS)**, Volume 5 Issue 2

Publisher: ACM Press

Full text available: pdf(2.69 MB) Additional Information: full citation, abstract, references, citings, index terms

A locking protocol to coordinate access to a distributed database and to maintain system consistency throughout normal and abnormal conditions is presented. The proposed protocol is robust in the face of crashes of any participating site, as well as communication failures. Recovery from any number of failures during normal operation or any of the recovery stages is supported. Recovery is done in such a way that maximum forward progress is achieved by the recovery procedures. Integration of ...

Keywords: concurrency, consistency, crash recovery, distributed databases, locking protocol

19 A protocol for failure and recovery detection to support partitioned operation in distributed database systems

Jung K. Kim, Geneva G. Belford

November 1986 **Proceedings of 1986 ACM Fall joint computer conference ACM '86**

Publisher: IEEE Computer Society Press

Full text available: pdf(957.95 KB) Additional Information: full citation, references, index terms

20 Investigating the use of analysis contracts to support fault isolation in object oriented code

L. C. Briand, Y. Labiche, H. Sun

July 2002 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 2002 ACM SIGSOFT international symposium on Software testing and analysis ISSTA '02**, Volume 27 Issue 4

Publisher: ACM Press

Full text available: pdf(574.83 KB) Additional Information: full citation, abstract, references, citings

A number of activities involved in testing software are known to be difficult and time consuming. Among them is the isolation of faults once failures have been detected. In this paper, we investigate how the instrumentation of contracts could address this issue. Contracts are known to be a useful technique to specify the precondition and postcondition of operations and class invariants, thus making the definition of object-oriented analysis or design elements more precise. Our aim in this paper ...

Keywords: contracts, object-oriented analysis, object-oriented testing, testability

Terms used **programming with assertions** Found **62,151** of **203,282**

Sort results by relevance Save results to a Binder

Display results expanded form Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown Relevance scale

1 Towards a method of programming with assertions

David S. Rosenblum

June 1992 **Proceedings of the 14th international conference on Software engineering ICSE '92**

Publisher: ACM Press

Full text available: pdf(1.16 MB) Additional Information: full citation, references, citings, index terms

2 Assertions and APL programming

Susan L. Gerhart

June 1975 **Proceedings of seventh international conference on APL APL '75**

Publisher: ACM Press

Full text available: pdf(576.09 KB) Additional Information: full citation, abstract, references, citings, index terms

Some difficulties in reading and writing APL programs are discussed. Assertions (important relations among and properties of variables) are suggested as a technique for improving reliability and readability of APL programs. It is shown how such assertions may be used for documentation, verification, optimization, and debugging. An implementation of run-time assertions in APLSV is given and illustrated by examples.

3 Programming languages (PL): Use of correctness assertions in declarative diagnosis

Lunjin Lu

March 2005 **Proceedings of the 2005 ACM symposium on Applied computing SAC '05**

Publisher: ACM Press

Full text available: pdf(210.69 KB) Additional Information: full citation, abstract, references

We use assertions to reduce the quantity of queries in declarative diagnosis of logic programs. We first present a declarative diagnoser for normal logic programs. Given a bug symptom, the diagnoser first constructs a tree that models the execution of the bug symptom and then searches the tree for the bug that causes the bug symptom. We then incorporate into the diagnoser three tree transformations that prune the tree before it is searched. These transformations make use of two kinds of assertio ...

Keywords: correctness assertions, declarative diagnosis, logic programs

4 CAD: Assertion-based automated functional vectors generation using constraint logic programming

Tun Li, Yang Guo, Si-Kun Li

April 2004 **Proceedings of the 14th ACM Great Lakes symposium on VLSI GLSVLSI '04**

Publisher: ACM Press

Full text available: pdf(275.33 KB) Additional Information: full citation, abstract, references, index terms

We present a novel approach to generate functional vectors based on assertions for RTL design verification. Our approach combines program-slicing based design extraction, word-level SAT and dynamic searching techniques. Through design extraction, vectors generation need only concern about the design parts related to the given assertion, thus large practical designs can be handled. Constraints Logic Programming (CLP) naturally models mixed bit-level and word-level constraints, and word-level SAT ...

Keywords: assertion, constraint logic programming, decision diagrams, functional verification, test generation

5 Assertions in programming languages

Richard N. Taylor

January 1980 **ACM SIGPLAN Notices**, Volume 15 Issue 1

Publisher: ACM Press

Full text available: pdf(768.24 KB) Additional Information: full citation, abstract, references, citings

The notion of embedding assertions in applications programs to aid in program verification and testing is not at all new; yet programming language designers seem loath to provide them, at least in useful ways. The Department of Defense language Ada is a case in point. The use of assertions is briefly reviewed, suggestions for their incorporation in languages is given, and an example of how they have been provided for the language HAL/S is shown.

6 Reports: A historical perspective on runtime assertion checking in software development

Lori A. Clarke, David S. Rosenblum

May 2006 **ACM SIGSOFT Software Engineering Notes**, Volume 31 Issue 3

Publisher: ACM Press

Full text available: pdf(473.68 KB) Additional Information: full citation, abstract, references, index terms

This report presents initial results in the area of software testing and analysis produced as part of the Software Engineering Impact Project. The report describes the historical development of runtime assertion checking, including a description of the origins of and significant features associated with assertion checking mechanisms, and initial findings about current industrial use. A future report will provide a more comprehensive assessment of development practice, for which we invite readers ...

7 Assertive comments in APL programming

David M. Weintraub

May 1986 **ACM SIGAPL APL Quote Quad , Proceedings of the international conference on APL APL '86**, Volume 16 Issue 4

Publisher: ACM Press

Full text available: pdf(234.42 KB) Additional Information: full citation, abstract, references, citings, index terms

Current practice in APL programming (and in most other languages) is to use comments to describe what is intended by subsequent lines within a function. An alternative approach to commenting is described, wherein assertions (a concept borrowed from program proof techniques) are inserted into APL code, specifying the present state of

relevant variables, etc. It is shown that this approach is especially appropriate for APL, as most APL design is accomplished by working from a desired end resu ...

8 On the theory of programming logics

Robert L. Constable

May 1977 **Proceedings of the ninth annual ACM symposium on Theory of computing STOC '77**

Publisher: ACM Press

Full text available: pdf(1.09 MB) Additional Information: full citation, abstract, references, citings, index terms

A new logic for reasoning about programs is proposed here, and its metamathematics is investigated. No new primitive notions are needed for the logic beyond those used in elementary programming and mathematics, yet the combination of these notions is remarkably powerful. The logic includes a programming language, designed with Michael O'Donnell, for program verification. It forms the core of the PL/CV verifier at Cornell. This study belongs to the discipline of Algorithmic Logic as conceive ...

9 Issues in software development: Harnessing curiosity to increase correctness in end-user programming

Aaron Wilson, Margaret Burnett, Laura Beckwith, Orion Granatir, Ledah Casburn, Curtis Cook, Mike Durham, Gregg Rothermel

April 2003 **Proceedings of the SIGCHI conference on Human factors in computing systems CHI '03**

Publisher: ACM Press

Full text available: pdf(318.50 KB) Additional Information: full citation, abstract, references, citings, index terms

Despite their ability to help with program correctness, assertions have been notoriously unpopular--even with professional programmers. End-user programmers seem even less likely to appreciate the value of assertions; yet end-user programs suffer from serious correctness problems that assertions could help detect. This leads to the following question: can end users be enticed to enter assertions? To investigate this question, we have devised a curiosity-centered approach to eliciting assertions ...

Keywords: assertions, curiosity, end-user software engineering, forms/3, surprise-explain-reward strategy

10 ASAP—a simple assertion pre-processor

Igor D.D. Curcio

December 1998 **ACM SIGPLAN Notices**, Volume 33 Issue 12

Publisher: ACM Press

Full text available: pdf(803.78 KB) Additional Information: full citation, abstract, index terms

Assertions are widely known as a powerful tool to detect software faults during the debugging of software systems. Despite the maturity of software engineering tools, assertions are seldom used in practice. ASAP is a pre-processor for C programs which implements several concepts defmed in the theory of formal specification, such as preconditions, postconditions, assertions related to intermediate states, loop invariants and variants, existential and universal quantifiers. In this paper, the noti ...

Keywords: assertions, pre-processor, programming techniques, software contract, software engineering

11 Axiomatic Definitions of Programming Languages: A Theoretical Assessment

Albert R. Meyer, Joseph Y. Halpern
April 1982 **Journal of the ACM (JACM)**, Volume 29 Issue 2
Publisher: ACM Press
Full text available: pdf(1.18 MB) Additional Information: full citation, references, citings, index terms

12 High level programming for distributed computing
Jerome A. Feldman
June 1979 **Communications of the ACM**, Volume 22 Issue 6
Publisher: ACM Press
Full text available: pdf(1.78 MB) Additional Information: full citation, abstract, references, citings

Programming for distributed and other loosely coupled systems is a problem of growing interest. This paper describes an approach to distributed computing at the level of general purpose programming languages. Based on primitive notions of module, message, and transaction key, the methodology is shown to be independent of particular languages and machines. It appears to be useful for programming a wide range of tasks. This is part of an ambitious program of development in advanced programmin ...

Keywords: assertions, distributed computing, messages, modules

13 First order programming logic
Robert Cartwright, John McCarthy
January 1979 **Proceedings of the 6th ACM SIGACT-SIGPLAN symposium on Principles of programming languages POPL '79**
Publisher: ACM Press
Full text available: pdf(937.15 KB) Additional Information: full citation, abstract, references, citings

First Order Programming Logic is a simple, yet powerful formal system for reasoning about recursive programs. In its simplest form, it has one major limitation: it cannot establish any property of the least fixed point of a recursive program which is false for some other fixed point. To rectify this weakness, we present two intuitively distinct approaches to strengthening First Order Programming Logic and prove that either extension makes the logic relatively complete. In the process, we prove t ...

14 Technical papers: empirical studies I: End-user software engineering with assertions in the spreadsheet paradigm
Margaret Burnett, Curtis Cook, Omkar Pendse, Gregg Rothermel, Jay Summet, Chris Wallace
May 2003 **Proceedings of the 25th International Conference on Software Engineering ICSE '03**
Publisher: IEEE Computer Society
Full text available: pdf(1.72 MB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

There has been little research on end-user program development beyond the activity of programming. Devising ways to address additional activities related to end-user program development may be critical, however, because research shows that a large proportion of the programs written by end users contain faults. Toward this end, we have been working on ways to provide formal "software engineering" methodologies to end-user programmers. This paper describes an approach we have developed for support ...

15 And/Or Programs: A New Approach to Structured Programming
David Harel
January 1980 **ACM Transactions on Programming Languages and Systems (TOPLAS)**,

Volume 2 Issue 1
Publisher: ACM Press

Full text available: pdf(1.04 MB) Additional Information: full citation, abstract, references, citings, index terms

A simple tree-like programming/specification language is presented. The central idea is the dividing of conventional programming constructs into the two classes of and and or subgoaling, the subgoal tree itself constituting the program. Programs written in the language can, in general, be both nondeterministic and parallel. The syntax and semantics of the language are defined, a method for verifying programs written in it is described, and the practical sig ...

16 Session 6: test automation: Performance assertions for mobile devices

Raimondas Lencevicius, Edu Metz
July 2006 **Proceedings of the 2006 international symposium on Software testing and analysis ISSTA '06**
Publisher: ACM Press
Full text available: pdf(266.70 KB) Additional Information: full citation, abstract, references, index terms

Assertions have long been used to validate the functionality of software systems. Researchers and practitioners have extended them for validation of non-functional requirements, such as performance. This paper presents the implementation and application of the performance assertions in mobile device software. When applying performance assertions for such systems, we have discovered and resolved a number of issues in assertion specification, matching, and evaluation that were unresolved in previo ...

Keywords: assertions, mobile devices, performance

17 Functional programming in C++ using the FC++ library

Brian McNamara, Yannis Smaragdakis
April 2001 **ACM SIGPLAN Notices**, Volume 36 Issue 4
Publisher: ACM Press
Full text available: pdf(553.51 KB) Additional Information: full citation, citings, index terms

18 Logic programming in APL

Robert Jernigan
June 1984 **ACM SIGAPL APL Quote Quad , Proceedings of the international conference on APL APL '84**, Volume 14 Issue 4
Publisher: ACM Press
Full text available: pdf(477.35 KB) Additional Information: full citation, abstract, references, citings, index terms

The programming of an expert system requires a language for specifying the rules that an expert uses, a data base for storing his knowledge, and a suitable interactive system. Logic programming has been described as a way of implementing expert systems. With logic programming, rules are expressed as assertions of what is true when certain conditions are true. To be true, the assertion has to be based upon fact or upon an inference derived from facts. This paper describes an implementation o ...

19 Concepts and Notations for Concurrent Programming

Gregory R. Andrews, Fred B. Schneider
March 1983 **ACM Computing Surveys (CSUR)**, Volume 15 Issue 1
Publisher: ACM Press

Full text available: pdf(4.02 MB) Additional Information: full citation, references, citings, index terms

20 Specifying programming language semantics: a tutorial and critique of a paper by Hoare and Lauer

I. Greif, A. Meyer

January 1979 **Proceedings of the 6th ACM SIGACT-SIGPLAN symposium on Principles of programming languages POPL '79**

Publisher: ACM Press

Full text available: pdf(817.79 KB) Additional Information: full citation, abstract, references, citings

Hoare and Lauer [1974] have advocated using a variety of styles of programming language definitions to fit the variety of users from implementers to program verifiers. They consider the question of whether different definitions and specifications determine the same language by showing that the definitions are what they call "consistent". However, their treatment skirts the question of whether their definitions can each be taken to specify the language adequately. Although, as we will show, any o ...

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next